In the United States Patent and Trademark Office
Board of Patent Appeals and Interferences

Appeal Brief

In re the Application of:

Duane Mark BALDWIN, Zhengwen HE, Allen Robert HEITMAN,
Hans Hanhsia LIN, and David Lynn MERBACH
Serial No. 09/972,010
Filed: October 5, 2001
Attorney Docket No. SJO920010093US1

STORAGE AREA NETWORK METHODS AND APPARATUS FOR COMMUNICATION AND INTERFACING WITH MULTIPLE PLATFORMS

Submitted by:

Konrad, Raynes & Victor LLP
315 So. Beverly Dr., Ste. 210
Beverly Hills CA 90212
(310) 556-7983
(310) 556-7984 (fax)

TABLE OF CONTENTS

I. Real Party in Interest

The entire right, title and interest in this patent application is assigned to real party in interest International Business Machines Corporation.

II. Related Appeals, Interferences, and Judicial Proceedings

Appellant, Appellant's legal representative, and Assignee are not aware of any other prior or pending appeals, interferences, and judicial proceedings which may be related to, directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

III. Status of the Claims

Claims 1, 3-21, and 23-26 are pending and have been rejected.

Claims 2 and 22 are canceled.

The final rejection dated July 14, 2006 ("Final Office Action") of the claims is being appealed for all pending claims 1, 3-21 and 23-26.

IV. Status of Amendments

Claims 14, 17, 18, and 24 were amended after receipt of the Final Office Action in the Response to Final Office Action dated October 6, 2006 ("Response FOA"). In the Advisory Action dated October 30, 2006, the Examiner declined to enter these amendments. Thus, the claims involved in the appeal will not include the amendments made in the Response FOA.

V. Summary of the Claimed Subject Matter

A. Independent Claim 1

Independent claim 1 is directed to a storage area network (SAN) including a first and second digital data processors executing a first and second operating systems, respectively, in communication with one or more storage devices. The Specification discloses a SAN having hosts, as shown in FIG. 43, that can have a variety of platforms, such as Windows, Solaris, AIX, etc. Per FIG. 1, the hosts communicate with storage devices. (FIG. 43, Specification, pgs. 42-43, 51, and 190-191)

The claim requires a first platform-specific process executing on the first digital data processor and a second platform-specific process executing on the second digital data processor, wherein the second operating system is different from the first operating system. With respect to this requirement, FIG. 43 discloses platform-specific processes in the form of the "platform specific scanners" that are for different operating systems, e.g., NT Scanner, AIX Scanner, etc. (Specification, pgs. 191-192)

The claim further requires a first common platform-independent process executing on the first digital data processor, wherein the first common platform-independent process invokes and communicates with a first command line interface of the first operating system to effect execution of the first platform-specific process via command line parameters and a second common platform-independent process executing on the second digital data processor, wherein the second common platform-independent process invokes and communicates with a second command line interface of the second operating system to effect execution of the second platform-specific process via command line parameters. With respect to these requirements, the Specification discloses that platform independent processes comprise portions of agents common to all platforms, shown as "common code" in FIG. 43 on different the different hosts having different operating systems. (Specification, pgs. 191-192). The Specification further discloses that the platform independent functions invoke and communicate with the platform dependent function via a command line interface. (Specification, pgs 42-43 and 193, lines 4-10)

B. Independent Claim 15

Independent claim 15 is directed a storage area network (SAN) including a first and second digital data processors executing a first and second operating systems, respectively, in communication with one or more storage devices. The Specification discloses a SAN having hosts, as shown in FIG. 43, that can have a variety of platforms, such as Windows, Solaris, AIX, etc. Per FIG. 1, the hosts are in communication with storage devices. (FIG. 43, Specification, pgs. 42-43, 51, and 190-191)

The claim requires a manager in communication with the SAN components. With respect to this requirement, FIG. 43 shows a SAN manager 20 that process and generates agent specific data. (Specification, pg. 190)

The claim requires a first platform-specific process executing on the first digital data processor and a second platform-specific process executing on the second digital data processor, wherein the second operating system is different from the first operating system. With respect to this requirement, FIG. 43 discloses platform-specific processes in the form of the "platform specific scanners" that are for different operating systems, e.g., NT Scanner, AIX Scanner, etc. (Specification, pgs. 191-192)

The claim further requires a first common platform-independent process executing on the first digital data processor, wherein the first common platform-independent process invokes and communicates with a first command line interface of the first operating system to effect execution of the first platform-specific process via command line parameters and a second common platform-independent process executing on the second digital data processor, wherein the second common platform-independent process invokes and communicates with a second command line interface of the second operating system to effect execution of the second platform-specific process via command line parameters. With respect to this requirement, the Specification discloses that platform independent processes comprise portions of agents common to all platforms, shown as "common code" in FIG. 43 on different the different hosts having different operating systems. (Specification, pgs. 191-192). The Specification further discloses that the platform independent functions invoke and communicate with the platform dependent function via a command line interface. (Specification, pgs 42-43 and 193, lines 4-10)

The claim further requires that the manager transmits a query to the first and second common platform-independent processes to request information regarding one or more of the SAN components and the platform independent processes invoke the first and second platform-specific processes, respectively, to obtain the requested information. With respect to this claim requirement, the Specification discloses, referring to FIGURE 43, the SAN manager 20 includes a service 510 which provides a communication interface for query engine 46 (of FIGURE 6). More specifically, service 510 transmits

and receives XML data to/from the agents 24. The query engine 46 coordinates running of the scanners (platform independent processes) and returning information to the client. A portion of the query engine 46 includes outband scanners which perform Simple Topology and Topology scans. (Specification, pgs. 63, 166) The Specification further discloses that the manager includes a query engine for forwarding queries to the platform independent processes, and the platform independent processors are invoked to obtain information, e.g., scans, regarding the status of the SAN components. (Specification, pg. 43)

C. Independent Claim 21

Independent claim 21 is directed a computer readable media including code executed by a first and second digital data processors having first and second operating systems in communication with one or more storage devices. FIG. 43 shows hosts (first and second digital data processors) having different operating systems (e.g., NT, AIX, Solaris, etc.) as including code. (FIG. 43, Specification, pgs. 42-43, 51, and 190-191)

The claim requires a first platform-specific process executing on the first digital data processor and a second platform-specific process executing on the second digital data processor, wherein the second operating system is different from the first operating system. With respect to this requirement, FIG. 43 discloses platform-specific processes in the form of the "platform specific scanners" that are for different operating systems, e.g., NT Scanner, AIX Scanner, etc. (Specification, pgs. 191-192)

The claim further requires a first common platform-independent process executing on the first digital data processor, wherein the first common platform-independent process invokes and communicates with a first command line interface of the first operating system to effect execution of the first platform-specific process via command line parameters and a second common platform-independent process executing on the second digital data processor, wherein the second common platform-independent process invokes and communicates with a second command line interface of the second operating system to effect execution of the second platform-specific process via command line parameters. With respect to this requirement, the Specification discloses that platform independent processes comprise portions of agents common to all

platforms, shown as "common code" in FIG. 43 on different the different hosts having different operating systems. (Specification, pgs. 191-192). The Specification further discloses that the platform independent functions invoke and communicate with the platform dependent function via a command line interface. (Specification, pgs 42-43 and 193, lines 4-10)

VI. Grounds of Rejection to Be Reviewed on Appeal

A concise statement listing each ground of rejection presented for review is as follows:

A. Claims 1, 3-21, and 23-26 are rejected under 35 U.S.C. §103(a) as being unpatentable over Bates (U.S. Patent No. 6,977,927) in view of Phillips (U.S. Patent No. 5,321,828).

VII. Argument

A. Rejection Under 35 U.S.C. §103 as Obvious over Bates in view of Phillips

1. Claims 1, 3, 21, and 24

First off, Applicants submit that the combination of Bates and Phillips is improper because Phillips, which is directed to operations of an in-circuit emulator (ICE) for debugging microprocessors and hardware, is non-analogous art with respect to the claimed communication between platform-specific and platform-independent processors on different digital data processors in a storage area network (SAN).

The Federal Circuit has announced a two-step test for determining whether references are analogous art.

> In order to rely on a reference as a basis for rejection of the applicant's invention, the reference must either be in the field of the applicant's endeavor or, if not, then be reasonably pertinent to the particular problem with which the inventor was concerned. See In re Deminski, 796 F.2d 436, 442, 230 USPQ 313, 315 (Fed. Cir. 1986). Patent examination is necessarily conducted by hindsight, with complete knowledge of the applicant's invention, and the courts have recognized the subjective aspects of determining whether an inventor would reasonably be motivated to go to the field in which the examiner found the reference, in order to solve the problem confronting the inventor. We have reminded ourselves and the PTO that it is necessary to consider "the reality of the circumstances", In re Wood, 599 F.2d 1032, 1036, 202 USPQ 171, 174 (CCPA 1979) -- in other words, common sense -- in deciding in which fields a person of ordinary skill

would reasonably be expected to look for a solution to the problem facing the inventor.

In re Oetiker, 24 USPQ2d 1443, 1445-46 (Fed. Cir. 1992); See, also Manual of Patent Examination and Procedure (MPEP) Sec. 2141.01(a).

With respect to the first test, Applicants submit that Phillips is not in the field of endeavor of the claims. Phillips is directed to an in-circuit emulator (ICE) that is used to debug and develop microprocessors. According to the "Field of the Invention" of Phillips, Phillips "relates generally to microcomputer systems and more particularly to instruments that enable the development and debugging of the hardware and software in target machines by the emulation and control of the target CPU within the target environment". (Phillips, col. 1, lines 6-12).

The claims, on the other hand, are directed to communication between platform-specific and platform-independent processors having first and second operating systems, respectively. Phillips is not in the field of the endeavor of the claims because Philips concerns an emulator for debugging microprocessors, which is a different field of endeavor than communication between platform-specific and platform-independent processes on first and second digital data processors having first and second operating systems, respectively. Applicants submit that an in-circuit emulator used for debugging machines is in a different field of endeavor than the claims which concern first and second platform independent processes that invoke first and second command line interfaces to effect execution of first and second platform specific processes on first and second digital data processors, respectively.

With respect to the second test, Applicants submit that the in-circuit emulator of Phillips used to debug microprocessors is not reasonably pertinent to the particular problems with which the claims of the present application are concerned, which concern communication between platform-specific and platform independent processes on different digital data processors using first and second command line interfaces, respectively. Applicants submit that an inventor working on issues related to the claimed subject matter concerning communication between platform-independent and platform-specific processors using command line interfaces would not be motivated to turn to the

field of Phillips, which concerns the use of an in-circuit emulator to debug microprocessors.

Applicants further submit that the Examiner has not provided any grounds that refute Applicants specific facts as to why Philips is non-analogous for not being in the same field of endeavor and not pertinent.

For these reasons, Applicants submit that it is improper for the Examiner to modify Bates with the teachings of Phillips because Phillips is non-analogous art. Thus, the rejection should be overturned on these grounds.

Applicants further submit that even if one were to combine Phillips and Bates (which Applicants submit is improper for the reasons explained above), the cited combination still does not teach or suggest the requirements of claims 1 and 21 for the following reasons.

The Examiner cites to col. 23, lines 50-67 of Phillips as teaching the claim requirements of first and second common platform independent processes executing on first and second processors, wherein the first and second common platform independent processes invoke and communicate with a first and second command line interfaces, respectively, to effect execution of first and second platform specific processes, respectively. (Final Office Action, pg. 3) Applicants traverse.

The cited col. 23 of Phillips discusses GDB, a standard debugger that runs on the UNIX operating system and is used as a starting point to implement a source level debugger. The source code of GDB is converted to a format compatible with a Microsoft "C" compiler running on DOS. Certain standard functions are altered to call their equivalents in "C". The GDB DLL retains its command line interface and does not allow Windows applications to directly link to its modules. The debugger also has provisions for communication with other Windows applications via ASII strings.

With regard to a command line interface, the cited col. 12 of Phillips mentions that a debugger maintains a command line interface to perform other debugger functions. See also, Phillips, col. 22, lines 19-25; col. 23, lines 12-14; col. 24, lines 18-22; col. 26, lines 3-7. For instance, Phillips mentions that a low level control interface 21 is used to provide system administration of the ICE 10 (in-circuit emulator) (Phillips, col. 19, lines 60-66); that the low level control interface 21 allows several options to be performed

from the command line (Phillips, col. 22, lines 19-21); and that the source level debugger is invoked with a DOS command line interface (Phillips, col. 24, lines 18-25).

Although Phillips discusses the use of a command line interface to invoke a debugger to perform debugger related operations, there is no teaching or suggestion in the cited sections of Phillips of the claimed first and second platform-independent processes that invoke and communicate with first and second command line interfaces of first and second operating systems to effect execution of first and second platform specific processes, respectively. For instance, there is no teaching or suggestion that the cited debugger invoked with a command line interface is a platform dependent process invoked by platform independent processes. Nor is there any teaching of platform independent processes on different digital data processes executing first and second operating systems invoking the debugger, or claimed first and second platform specific processes.

Moreover, Applicants submit that Phillips teaches away from the claim requirements of first and second platform independent processes using first and second command line interfaces to effect execution of first and second platform specific processes because Phillips mentions that the debugger is invoked with DOS command line or a UNIX shell command (Phillips, col. 24, lines 17-25), which are platform specific processes, not platform independent processes. Further, the Examiner has not cited any part of Phillips that teaches or suggests that the cited command lines are invoked by platform independent processes to invoke platform specific processes as claimed.

In the Advisory Action (second point of contention), the Examiner restated his finding on pg. 14 of the Final Office Action that the "prior art teaches how GDB (Unix based) communicates with Windows applications via ASCII in command line windows, which are retained." (Final Office Action, pg. 14) Applicants traverse the relevance of this finding because the cited col. 23 mentions that the debugger (GDB) retains a command line interface. This mention of a debugger retaining a command line interface nowhere teaches, suggests or mentions first and second common platform independent processes executing on first and second processors separately invoking and communicating with first and second command line interfaces of first and second operating systems to effect execution of first and second platform specific processes as claimed.

Thus, even if one were to modify Bates with Phillips, Phillips discussion of the use of command line interfaces would only suggest that one may use command line interfaces in the systems of Bates to invoke a debugger. However, such proposed modification nowhere teaches or suggests the specific claimed use of command line interfaces, i.e., that first and second platform-independent processes invoke and communicate with first and second command line interfaces of first and second operating systems to effect execution of first and second platform specific processes, respectively.

Accordingly, Applicants request reversal of the rejection of claims 1 and 21 because the cited combination of art does not teach or suggest all the claim requirements.

Applicants reversal of claims 3 and 24 because these claims depend from claims 1 and 21, which are patentable over the cited art for the reasons discussed above.

2. Claims 15

Claim 15 substantially includes the requirements of claim 1 and additionally requires that manager transmits a query to the first and second common platform-independent processes to request information regarding one or more of the SAN components and the platform independent processes invoke the first and second platform-specific processes, respectively, to obtain the requested information.

The Examiner cited col. 13, line 29 to col. 14, line 60 of Bates as teaching these additional requirements of claim 15. (Final Office Action, pg. 4) Applicants traverse.

The cited cols. 13-14 of Bates mentions that a storage allocator maps or masks available storage space to present to hosts. The cited cols. 13-14 further mentions virtual LUN partitions and storage security. Each host, having different operating systems, has access to separate non-overlapping physical LUNs. The storage allocator may be controlled by a user interface to manually configure the allocation of storage. The storage allocator is implemented in a SAN appliance or device. Users may use a GUI to allocate storage using the storage allocator. Bates further mentions that the storage allocator receives I/O requests from servers, maps the data I/O requests to physical storage I/O requests and forwards them to storage. (Bates, col. 3, lines 37-41).

Although the cited storage allocator discusses how to present a storage space to hosts, nowhere does the cited Bates teach or suggest that the storage allocator transmits a

query to first and second platform independent processes on hosts (digital data processors) to request information regarding SAN component, where the platform independent processes invoke platform specific processes to obtain the requested information. The Examiner has not cited any part of Bates that teaches that the cited storage allocator transmits a query to platform independent processes to invoke platform specific processes to obtain information as claimed. Instead, the cited cols. 13-14 discuss how a storage allocator masks available storage space for hosts and provides the storage space to the hosts.

Accordingly, Applicants request reversal of the rejection of claim 15 because the cited art does not teach or suggest the claim requirements, including those requirements also found in the cited claims 1 and 21.

3. Claims 4, 5 and 23

Claims 4 and 23 are patentable over the cited art because they depend from claims 1 and 21. Moreover, these claims also provide additional grounds of patentability over the cited art for the following reasons.

Claims 4 and 23 depend from claims 1 and 21, respectively, and further require a manager in communication with the first and second common platform-independent process to transmit requests thereto for information regarding one or more components of the SAN.

The Examiner cited col. 13, line 29 to col. 14, line 60 of Bates as teaching the requirements of claims 4 and 23. (Final Office Action, pg. 5) Applicants traverse.

The cited cols. 13-14 of Bates mentions that a storage allocator maps or masks available storage space to present to hosts. The cited cols. 13-14 further mentions virtual LUN partitions and storage security. Each host, having different operating systems, has access to separate non-overlapping physical LUNs. The storage allocator may be controlled by a user interface to manually configure the allocation of storage. The storage allocator is implemented in a SAN appliance or device. Users may use a GUI to allocate storage using the storage allocator. Further, the storage allocator may automatically allocate storage by operation of an algorithm.

Bates further mentions that the storage allocator receives I/O requests from servers, maps the data I/O requests to physical storage I/O requests and forwards them to storage. (Bates, col. 3, lines 37-41). Bates also mentions that the storage allocator may be implemented in a multi-platform, platform independent, programming language. (Bates, col. 15, lines 5-21).

Nowhere do the cited cols. 13-14 anywhere teach or suggest a manager in communication with first and second common platform-independent process to transmit requests thereto for information regarding one or more components of the SAN. Instead, the cited cols. 13-14 discuss how a storage allocator implemented on a SAN appliance allocates storage space to hosts and interfaces between servers and storage. This does not disclose a manager transmitting requests to first and second common platform-independent processes as claimed. For instance, the Examiner has not cited any part of Bates teaching that the storage allocator submits requests to first and second platform independent processes that effect execution of first and second platform-specific processes, where the platform independent and platform specific processors execute on a same digital data processor.

Accordingly, Applicants reversal of the rejection of claims 4 and 23 because the cited art does not disclose the additional requirements of these claims.

Applicants reversal of claim 5 because this claim depends from claim 4, which is patentable over the cited art for the reasons discussed above.

4. Claim 16

Claim 16 is patentable over the cited art because it depends from claim 15, which is patentable over the cited art for the reasons discussed above. Moreover, claim 16 also provides additional grounds of patentability over the cited art for the following reasons.

Claim 16 depends from claim 15 and further requires that the invoked first or second platform specific process gathers information regarding one or more of the SAN components and transmits the information to the first or second command line interface of its respective digital data processor operating system.

The Examiner cited col. 23, lines 50-67 of Phillips as teaching the claim requirements of communication via a command line interface. (Final Office Action, pg. 6)

The cited col. 23 discusses the use of a command line interface to perform debugger operations. Nowhere does the cited col. 23 anywhere teach or suggest that first or second platform specific processes invoked via first or second platform independent processes, transmits gathered information on SAN components to first or second command line interfaces.

Accordingly, Applicants request reversal of the rejection of claim 16 because the cited art does not teach or suggest the additional requirements of claim 16.

5. Claims 6 and 25

Claims 6 and 25 are patentable over the cited art because they depend from claims 1 and 21, which are patentable over the cited art for the reasons discussed above. Moreover, these claims provide additional grounds of patentability over the cited art for the following reasons.

Claims 6 and 25 depend from claims 1 and 21, respectively, and further require that the invoked first and second platform specific processes gather information regarding one or more SAN components and transmit the information to the Standard Output/Error of their respective first and second digital data processors.

The Examiner cited col. 3, lines 37-67 of Bates and the previously discussed col. 8 of Bates and col. 23 of Phillips as teaching the additional requirements of claim 6. (Final Office Action, pg. 7) Applicants traverse.

The cited col. 3 discusses the above discussed network of servers with different operating systems, a storage allocator and storage, where the storage allocator receives read and write requests from the servers to determine the storage locations for the request. The discussed cited col. 23 of Phillips discusses a debugger that has a command line interface through which it may be invoked. Nowhere do the cited Phillips and Bates anywhere teach or suggest separate first and second platform specific processes executing on different processors having different operating systems gathering information on SAN components and transmitting the gathered information to the standard output/error.

Accordingly, Applicants request reversal of the rejection of claims 6 and 25 because the cited art does not teach or suggest the additional requirements of these claims.

6. Claim 7, 8, and 17

Claims 7, 8, and 17 are patentable over the cited art because they depend from intervening claims 6 and 16 and base claims 1 and 15, which are patentable over the cited art for the reason discussed above. Further, claims 7 and 17 provide additional grounds of patentability over the cited art for the following reasons.

Claim 7 and 17 depend from claims 6 and 16 and further require that .the first and second common platform independent processes capture the information in a Standard Output/Error transmitted by the invoked platform specific process.

The Examiner cited col. 3, lines 37-67 of Bates and the previously discussed and col. 23 of Phillips as teaching the additional requirements of claims 7 and 17. (Final Office Action, pg. 8) Applicants traverse.

The cited col. 3 of Bates discusses the above discussed network of servers with different operating systems, a storage allocator and storage, where the storage allocator receives read and write requests from the servers to determine the storage locations for the request. The discussed cited col. 23 of Phillips discusses a debugger that has a command line interface through which it may be invoked. Nowhere do the cited Phillips and Bates anywhere teach or suggest the claim requirements that separate first and second common platform independent processes capture information on SAN components in the Standard Output/Error transmitted by the invoked platform specific processes.

Accordingly, Applicants request reversal of the rejection of claims 7 and 17 because the cited art does not teach or suggest the additional requirements of these claims.

Applicants request reversal of the rejection of claim 8 because this claim depends from dependent claim 7 and base claim 1, which are patentable over the cited art for the reasons discussed above.

7. Claims 9, 10-14, 18, 19, 20, and 26

Claims 9, 18, and 26 are patentable over the cited art because they depend from base claims 1, 15, and 21, which are patentable over the cited art for the reason discussed above. Further, these claims provide additional grounds of patentability over the cited art for the following reasons.

Claims 9, 18, and 26 depend from claims 4, 17, and 24, respectively, and further recite that the manager comprises a query engine for transmitting the requests to the first and second common platform independent processes.

The Examiner cited col. 15, lines 5-22 of Bates as teaching a query and col. 3, lines 46-67 of Bates as teaching multiple processors and platform specific operations. (Final Office Action, pgs. 9-10) Applicants traverse.

The cited col. 15 mentions that the storage allocator is implemented in a platform independent language, such as Java. The cited col. 3 discusses how the storage allocator provides access to storage to servers having different operating systems. Although the cited Bates mentions that the storage allocator is in a platform independent language, nowhere does the cited Bates (or Phillips) anywhere teach or suggest that the storage allocator has a query engine for transmitting requests to first and second common platform independent processes on different processors having different operating systems as claimed. Further, FIG. 7 shows the storage allocator separate from the servers. The claims require that first and second platform independent processes receive requests from a query engine. Nowhere does the cited Bates anywhere teach or suggest that the servers 702, 704, and 706 in FIG. 7 have platform independent processes for receiving requests for information on SAN components from a query engine. Instead, the cited Bates discusses how the storage allocator receives read and write requests from the servers, where the storage allocator is written in a platform independent computer language.

Applicants request reversal of the rejection of claims 9, 18, and 26 because the cited art does not teach or suggest the additional requirements of these claims.

Applicants request reversal of the rejection of claims 10-14, 19, and 20 because these claims depend from intervening claims 9, 18, and 26 and base claims 1, 15, and 21, which are patentable over the cited art for the reasons discussed above

VIII. Conclusion

Each of the rejections set forth in the Final Office Action are improper and should be reversed.

Respectfully submitted,

/David Victor/

David W. Victor
Reg. No. 39,867

Dated: February 12, 2007

Direct All Correspondence to:
David Victor
Konrad Raynes & Victor LLP
315 South Beverly Drive, Ste. 210
Beverly Hills, California 90212
Tel: 310-553-7977
Fax: 310-556-7984

IX. Claims Appendix

1. (Previously Presented) A storage area network (SAN) including a first and second digital data processors executing a first and second operating systems, respectively, in communication with one or more storage devices, comprising:

a first platform-specific process executing on the first digital data processor;

a second platform-specific process executing on the second digital data processor, wherein the second operating system is different from the first operating system;

a first common platform-independent process executing on the first digital data processor, wherein the first common platform-independent process invokes and communicates with a first command line interface of the first operating system to effect execution of the first platform-specific process via command line parameters; and

a second common platform-independent process executing on the second digital data processor, wherein the second common platform-independent process invokes and communicates with a second command line interface of the second operating system to effect execution of the second platform-specific process via command line parameters.

2. (Canceled)

3. (Previously Presented) The SAN of claim 1, wherein each of the first and the second operating systems can be any of a Unix™, a Windows™, Solaris, AIX operating systems.

4. (Previously Presented) The SAN of claim 1, further comprising a manager in communication with the first and second common platform-independent processes to transmit requests thereto for information regarding one or more components of the SAN.

5. (Previously Presented)The SAN of claim 4, wherein the first and second common platform independent processes respond to the requests from the manager by invoking the first and second platform-specific processes, respectively.

6. (Previously Presented)The SAN of claim 5, wherein the invoked first and second platform specific processes gather information regarding one or more SAN components and transmit the information to the Standard Output/Error of their respective first and second digital data processors.

7. (Previously Presented)The SAN of claim 6, wherein the first and second common platform independent processes capture information in the Standard Output/Error transmitted by the invoked first and second platform specific process, respectively.

8. (Previously Presented)The SAN of claim 7, wherein the first and second common platform independent processes transmit the captured information to the manager for further processing.

9. (Previously Presented)The SAN of claim 4, wherein the manager comprises a query engine for transmitting the requests to the first and second common platform independent processes.

10. (Previously Presented) The SAN of claim 9, wherein the query engine comprises a registry identifying the first and second common platform independent processes and the first and second digital data processors, respectively, associated therewith.

11. (Previously Presented) The SAN of claim 10, wherein the registry provides one or more identifiers for communicating with the first and second common platform independent processes.

12. (Previously Presented) The SAN of claim 9, wherein the query engine formats the request in a mark-up language format.

13. (Previously Presented) The SAN of claim 12, wherein the mark-up language can be any of XML and HTML.

14. (Previously Presented) The SAN of claim 13, wherein the first and second platform independent processes format the captured information in a mark-up language format for transmission to the manager.

15. (Previously Presented) A storage area network having first and second digital data processors and one or more storage devices in communication with the digital data processors, comprising:

a manager in communication with the SAN components;

a first platform-specific process executing on the first digital data processor, the first digital data processor executing under a first operating system;

a second platform-specific process executing on the second digital data processor, the second digital data processor executing under a second operating system different from the first platform;

a first common platform-independent process executing on the first digital data processor, wherein the first common platform independent process invokes and communicates with a first command line interface of the first operating system to effect execution of the first platform-specific process via one or more command-line parameters;

a second common platform-independent process executing on the second digital data processor, wherein the second common platform independent process invokes and communicates with a second command line interface of the second operating system to effect execution of the second platform-specific process via one or more command-line parameters; and

the manager transmits a query to the first and second common platform-independent processes to request information regarding one or more of the SAN components and the platform independent processes invoke the first and second platform-specific processes, respectively, to obtain the requested information.

16. (Previously Presented) The SAN of claim 15, wherein the invoked first or second platform specific process gathers information regarding one or more of the SAN components and transmits the information to the first or second command line interface of its respective digital data processor operating system.

17. (Previously Presented) The SAN of claim 16, wherein the first and second common platform independent processes capture the information in a Standard Output/Error transmitted by the invoked platform specific process.

18. (Previously Presented) The SAN of claim 17, wherein the manager comprises a query engine for forwarding the query from the manager to the first and second common platform independent process.

19. (Previously Presented) The SAN of claim 18, wherein the query engine comprises a registry containing information for identifying the first and second common platform independent processes and their respective first and second digital data processors.

20. (Previously Presented) The SAN of claim 19, wherein the first and second common platform independent processes register with the registry to provide identification information thereto.

21. (Previously Presented) Computer readable media including code executed by a first and second digital data processors having first and second operating systems in communication with one or more storage devices, wherein the code comprises:

a first platform-specific process executing on the first digital data processor;

a second platform-specific process executing on the second digital data processor, wherein the second operating system is different from the first operating system;

a first common platform-independent process executing on the first digital data processor, wherein the first common platform-independent process invokes and

communicates with a first command line interface of the first operating system to effect execution of the first platform-specific process via command line parameters; and

a second common platform-independent process executing on the second digital data processor, wherein the second common platform-independent process invokes and communicates with a second command line interface of the second operating system to effect execution of the second platform-specific process via command line parameters.

22. (Canceled)

23. (Previously Presented) The computer readable media of claim 21, further comprising:

a manager in communication with the first and second common platform-independent processes to transmit requests thereto for information regarding one or more components of the SAN.

24. (Previously Presented) The computer readable media of claim 21, wherein the first and second common platform independent processes respond to the requests from the manager by invoking the first and second platform-specific processes, respectively.

25. (Previously Presented) The computer readable media of claim 24, wherein the invoked first and second platform specific processes gather information regarding one or more SAN components and transmit the information to the Standard Output/Error of their respective first and second digital data processors.

26. (Previously Presented) The computer readable media of claim 24, wherein the manager comprises a query engine for transmitting the request to the first and second common platform independent processes.

X. Evidence Appendix

None

XI. Related Proceedings Appendix

None